REMARKS

1. New Claims

Applicant adds a new independent Claim 20, which is identical to independent Claim 12, but which has as an additional limitation the following paragraph:

> "a decoder for decoding the code words of the first set and the code words of the second set."

Furthermore, Applicant adds an additional independent Claim 21, which is identical to method Claim 19, but which includes the following paragraph:

> "decoding the code words of the first set and the code words of the second set."

These amendments are supported by several passages of the translation of the PCT application as originally filed, such as page 2, line 6, page 6, lines 29-30, page 7, lines 12-14, page 7, line 38, page 8, line 9, page 8, line 20, page 9, line 30, page 11, lines 4 and 9, page 17, line 1, page 18, line 38 to page 19, line 1 or page 19, line 24.

2. 35 USC 101

Claims 12 and 19 do indeed, provide a useful, concrete and tangible result, as discussed below. Accordingly, Applicant respectfully traverses the Examiner's rejection thereof under 35 USC 101.

As defined in the first paragraph of Claim 12 or the first paragraph of Claim 19, the input into the apparatus for reading a data stream is a data stream in which code words of a first set and code words of a second set are included in a specific order, *i.e.* code words of the first set start at the corresponding first raster

points of the segments and code words of the second set start at corresponding second raster points of the corresponding segments, while the writing direction is reversed for the first and the second sets. Importantly, the data stream is particularly written such that a code word of the second set is distributed over more than one segment in accordance with a predetermined rule. Thus, while this data stream includes a lot of bits, when one simply reads this data stream bit by bit in a single reading direction the data stream is completely useless.

However, when the data stream is processed by the inventive device, then one has retrieved the code words from the first set and the code words of the second set, although they were written in the data stream with different directions and the code word of the second set was distributed over two or more segments. Thus, although the data stream input into the inventive device of Claim 12 was useless, the code words output by the device of Claim 12 are useful, because they represent the message which is encoded by the code words in an understandable and interpretable manner.

Thus, the results of the inventive reading device are the code words of the first set and the code words of the second set, which are now retrieved as they are and are, as such, useful, concrete and tangible. One could, for example, store these code words without any specific reordering so that these code words can later be retrieved for a straight-forward decoding operation. Alternatively, one could as well decode these code words as defined by new independent Claims 20 and 21. Both cases are important and represent useful application due to the fact that the specific ordering of the code words in the data stream input into the inventive reader has a specific error resilience or error robust characteristic, on the one hand, and a high efficiency on the other hand, because the data stream does not have to have any free places. Instead, in view of the fact that a separation of a code word into different parts is permissible, all bit positions of a data stream can now be fully used, irrespective of the length of the different code words which can be, for example, code words having variable lengths.

Thus, as stated above, the input into the inventive reader is a data stream, which can not be simply treated by a straight-forward decoder, because the data stream is different from a straight-forward data stream, in which one code word is written after the other in the same direction and in which a separation of code words over different segments is not allowed.

The concrete, useful and tangible result of the inventive reader is the specific code words retrieved from the data stream, which can now be used for storing or decoding via a straight-forward decoder accepting one code word after the other.

Finally, note that the data stream ultimately specifies a raster, *i.e.* a useful, concrete and tangible representation of the data stream.

3. 35 USC 102

In the response to amendment section 1 on page 2 of the Office Action, the Examiner unfortunately fully ignored the following features of Applicant's Claims 12 and 19:

- a code word of the second set is distributed over more than one segment in accordance with a predetermined rule;

- the control device is adapted for jumping at least to a further segment different from the segment, in which the part of the code word of the second set has been found, in accordance with the predetermined rule, when all segments have been searched for code words of the second set in accordance with the predetermined assignment rule and at least only a part of the code word of the second set has been found in a segment and the code word of the second set is still not complete; and

- the control device is adapted for obtaining the at least one code word of the second set completely or a further part of the at least one code word of the second set from the further segment.

To summarize, the Examiner unfortunately ignored all key limitations introduced in the recent claim amendments.

In section 7 of the Office Action, the Examiner goes into more detail. Before discussing the Examiner's remarks in detail, Applicant summarizes the features of Nagai, which are pertinent to the invention.

Fig. 2A, Fig. 3A or Fig. 4A illustrate a separation of a picture into 35 macro blocks, which are separately processed by a video encoding system in accordance with the MPEG standard. Please refer to column 1, line 48, where these blocks are called "regions". In Fig. 26, these regions are called macro blocks, as outlined, for example, in column 22, lines 21, 24, 25 etc. As clearly illustrated in Fig. 26 and also in Fig. 6, a "code string" in Nagai always has a number of code words of variable length or fixed length as, for example, described in column 21, lines 5 and 6. Because, as for example illustrated in Fig. 26, 27, 28, or 29, a code string includes a complete macro block MB, which means that each code string only has complete code words. Thus, Nagai does not show the limitation that "a code word of the second set is distributed over more than one segment in accordance with a predetermined rule" (see first paragraph of Claim 12). Interestingly, the Examiner also mentions this limitation on page 4, lines 3, 4 and 5 of the Office Action. The Examiner did not give any reference for this limitation in Nagai.

In the second paragraph of page 5, the Examiner tries to find the control device in Nagai. To this end, the Examiner refers to column 25, line 44 to column 26, line 34, in which the device of Fig. 36 is discussed. The device illustrated in Fig. 36 works as follows:

As illustrated in column 5, lines 33 and 34a, synchronizing code detection section 105 detects two subsequent synchronizing codes, and a buffer 106 stores the coding data between adjacent synchronizing codes. What happens then is illustrated in Fig. 33A. From the start of the synchronizing code, a decoding in the forward (order) direction is performed and, additionally, a decoding operation in the reverse (inverse) direction is performed from the end of the stored data. When one interprets the left vertical line in the Figs. 37A to D as a first point of a segment and the right horizontal lines in these figures as the end of a segment, these figures illustrate one segment, and when this is considered together with column 25, line 34, the procedure in Nagai is that one segment is processed after the other. This means that a segment is stored, processed in forward and reverse direction and then it is decided which part is used from the forward direction operation and the backward direction operation and which part of the data is discarded. Then, the procedure of this segment is completed and the next segment is processed in an identical way.

Therefore, when in a first segment, for example, of Fig. 37A, only a part of the second code word is found, an error is output because the coding tree corresponding to this part of the code word of the second set in the current segment cannot readily be completed. This means that the error detection and error processing capabilities of Nagai for this incomplete code word of the second set is performed. Importantly, there is no processing over more than one segment, but Nagai simply discloses to process one segment after the other, as clearly disclosed in column 25, lines 18 to 23.

Thus, regarding the penultimate paragraph of Claim 12, the limitation that the control device is adapted for jumping at least to a further segment different from the segment, in which the part of the code ward of the second set is found is not fulfilled because Nagai would never recognize that some bits are a part of a code word of the second set. Instead, when this happens, Nagai would output an error.

Thus, Nagai would also not process more than one segment in accordance with a predetermined rule, when a part of a code word of the second set has been found. Additionally, Nagai would not search for code words of the second set in several segments because only each segment is processed one after the other. Particularly, such a search in Nagai does not take place in accordance with the predetermined assignment rule.

Furthermore, only a part of the code word of the second set cannot be found in the segment because Nagai would, in such a case, simply output an error. This is what happens in Nagai when a code word is still not complete.

Then, regarding the last paragraph of Claim 12, the at least one code word of the second set is completed by taking a further part of the at least one code word of the second set from a further segment, *i.e.* segment which is different from the segment where the first part was. Thus, parts of a single code word, which are in different segments are put together by the control device as defined in the last paragraph of Claim 12, which is not done in Nagai. As clearly stated in column 25, lines 44 to 50, and as illustrated by Figs. 36 and 37A-D, Nagai teaches to perform a segment by segment processing, wherein all decisions are done within a single segment and a combination of bits from two segments to obtain a code word distributed over different segments is not done at all.

Thus, the Examiner is not correct in stating that Nagai anticipates Claim 12.

The Examiner states in the last line of section 7 on page 5 of the Office Action that "determining errors and deciding if error is discarded or used could correspond to the last paragraph of Claim 12". However, Nagai would never use the first part of a code word in a first segment and the second part of a code word in the second segment, since both parts would, in the device in Nagai, show up as errors because the forward or backward decoding trees would result in

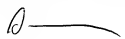
showing a fragment of a code word, *i.e.* the first portion or the second portion of part of the code word as an erroneous code word.

Thus, independent Claims 12 and 19 are not anticipated.

Regarding new Claims 20 and 21, I can see that there is even another difference. Because Nagai would recognize a code word distributed over several segments as errors in each segment, and because Nagai does not disclose to perform an inter-segment combination of parts of a code word, Nagai cannot teach the step of decoding such a code word of the second set. Instead, such a code word of the second set cannot be decoded by Nagai because Nagai would show an error for each segment, in which a part of this code word of the second set is included.

Should the Examiner deem it helpful, he is encouraged to contact Applicant's attorney, Michael A. Glenn at (650) 474-8400.

Respectfully submitted,



Michael A. Glenn
Reg. No. 30,176

Customer No. 22,862